

DESIGN IDEAS

With each iteration, the program checks the network resistance R_{EQ} at the low, middle, and high temperatures. It seeks to achieve an equal resistance change in the upper and lower halves of the temperature range by trying all combinations of R and RS by substituting values from the data statements. It prints a result, searches for a better result, prints that, and continues

(Fig 2). Remember to denormalize the final result by multiplying the resistance values by 1k. Typical execution time is about 10 minutes. **EDN**

EDN

To Vote For This Design, Circle No 748

```

INPUT LOW TEMPERATURE: 20
INPUT LOW TEMP RESISTANCE: 127
INPUT HIGH TEMPERATURE: 130
INPUT HIGH TEMPERATURE RESISTANCE: 2.64
THE CALCULATED MID-POINT TEMPERATURE = 75 DEGREES.
INPUT THERMISTOR RESISTANCE AT 75 DEGREES: 12.96
*****

BEST SO FAR:
RS = 1 OHMS, AND R = 1 OHMS.
   *** (AT 20 DEGREES,  $R_{EQ}$  = .9922482 OHMS.)
   *** (AT 75 DEGREES,  $R_{EQ}$  = .9331552 OHMS.)
   *** (AT 130 DEGREES,  $R_{EQ}$  = .7844828 OHMS.)
DELTA1 = .1486723 OHMS, AND DELTA2 = .059093 OHMS.
THE TOTAL CHANGE IN RESISTANCE FROM 20 TO 130 DEGREES IS .2077653 OHMS.
*****
      *
      *
      *
(12 ITERATIONS)
      *
      *
      *
*****
Working...

BEST SO FAR:
RS = 8.66 OHMS, AND R = 1.07 OHMS.
   *** (AT 20 DEGREES,  $R_{EQ}$  = 1.061627 OHMS.)
   *** (AT 75 DEGREES,  $R_{EQ}$  = 1.019542 OHMS.)
   *** (AT 130 DEGREES,  $R_{EQ}$  = .9774454 OHMS.)
DELTA1 = .0420962 OHMS, AND DELTA2 = .04208481 OHMS.
THE TOTAL CHANGE IN RESISTANCE FROM 20 TO 130 DEGREES IS .08418102 OHMS.
*****
Working...

END OF CALCULATIONS.
```

Fig 2—This example of Listing 1's output shows two of the 14 iterations that the program prints in determining the R_{EQ} network in Fig 1. $R=1.07\text{ k}\Omega$ and $RS=8.66\text{ k}\Omega$ are the final values.

Circuit converts 7-segment code to decimal

P Mazumder
University of Illinois, Urbana, IL

The 3-IC circuit in Fig 1 accepts 7-segment input code representing either of two commonly used numeric-character sets, converts it to an intermediate BCD code, and converts that to decimal code. The 7-segment

code, defined in **Fig 2**, can be active high or active low. In addition, for convenience sake you can substitute exclusive-NOR (XNOR) gates for the exclusive-OR (XOR) gates shown.

Table 1 shows the active-high output (P_i) that corresponds to each 7-segment character in the two sets, along with the intermediate BCD code (ABCD) that

Ah, there has one. But it can't afford a plotter. Which got there was reasonably reasonable to hook on a workstatic plotter. Present Plotter. For an engineer's perfect plot. Consider it.

It can
smooth an
character
drafting fi
using eig
Furth

DESIGN IDEAS

serves as the input to IC₃. Note that the BCD-to-decimal decoder IC₃ establishes a fixed relationship between ABCD and P_j, but for each character set a given character produces a different BCD code. The sets differ in their representation of "6" and "9": set #2 uses an extra segment for these characters.

Table 2 shows the connections you must make to implement each of the four possible sets of input code (character set #1, active high or low; character set #2, active high or low). You connect the signals for segments f and g to IC_{1B} (Fig 1) for all code sets. Each of the remaining inputs (IN₁ through IN₈) connect to 0 (GND), to 1 (V_{CC}), or to one of the segment signals, as shown in Table 2.

The XOR gates convert the applied 7-segment code to active-high BCD code. Table 3 depicts the logical composition of each bit in the BCD code for each of the four possible input-code sets, using either XOR gates (top half of Table 3) or XNOR gates (bottom half).

Note that if you build the circuit with XNOR gates, IN₃ is the only input affected; the reason is that you make the opposite connections to IN₃ (V_{CC} as opposed to GND). Also, note that you can eliminate any XOR gate for which one input is always low because the gate's output then follows the other input.

EDN

To Vote For This Design, Circle No 749

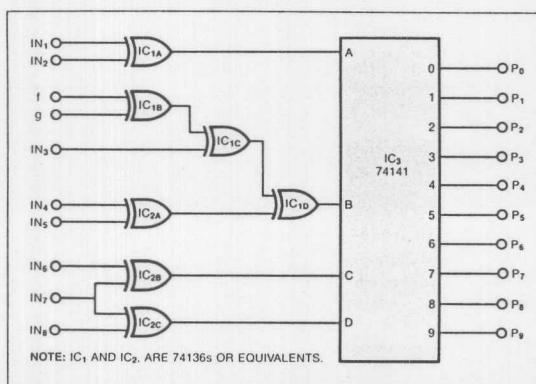


Fig 1—This 3-IC circuit converts 7-segment code (either active-high or active-low for either of two numeric character sets) to decimal code.

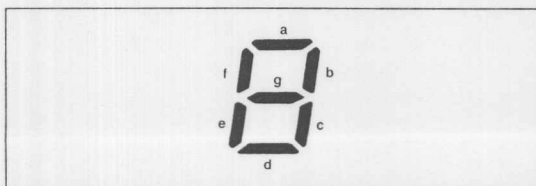


Fig 2—This standard character format lets you represent numbers by selectively illuminating the seven segments labeled a through g.

TABLE 1—CHARACTER SETS

CHARACTER SET #1	A	B	C	D	OUTPUT
0	0	0	0	0	P ₀
1	0	0	1	0	P ₄
2	1	0	0	0	P ₁
3	1	0	0	1	P ₉
4	1	0	1	0	P ₅
5	0	0	0	1	P ₈
6	1	1	0	0	P ₃
7	1	1	1	0	P ₇
8	0	1	0	0	P ₂
9	0	1	1	0	P ₆

CHARACTER SET #2	A	B	C	D	OUTPUT
0	1	1	0	0	P ₃
1	1	0	0	1	P ₉
2	0	1	0	0	P ₂
3	0	0	1	0	P ₄
4	0	0	0	1	P ₈
5	1	1	1	0	P ₇
6	1	0	0	0	P ₁
7	1	0	1	0	P ₅
8	0	0	0	0	P ₀
9	0	1	1	0	P ₆

TABLE 2—INPUT CODES

INPUT CODE	IN ₁	IN ₂	IN ₃	IN ₄	IN ₅	IN ₆	IN ₇	IN ₈	GATES REQUIRED
CHARACTER SET #1 ACTIVE HIGH	a	f	1(0)	a	b	1	d	e	7
CHARACTER SET #2 ACTIVE HIGH	b	g	0(1)	d	e	e	a	1	6
CHARACTER SET #1 ACTIVE LOW	a	f	1(0)	a	b	0	d	e	6
CHARACTER SET #2 ACTIVE LOW	b	g	0(1)	d	e	e	a	0	5

*INPUTS IN PARENTHESES ARE FOR USE WITH XNOR GATES ONLY.

1=LOGIC 1 (V_{CC}); 0=LOGIC 0 (GND).

TABLE 3—LOGICAL MAKEUP OF BCD CODE

INPUT CODE	A	B	C	D
CHARACTER SET #1 ACTIVE 1	a ⊕ f	((f ⊕ g) ⊕ 1) ⊕ (a ⊕ b)	d ⊕ 1	d ⊕ e
CHARACTER SET #2 ACTIVE 1	b ⊕ g	(f ⊕ g) ⊕ (d ⊕ e)	e ⊕ a	a ⊕ 1
CHARACTER SET #1 ACTIVE 0	a ⊕ f	((f ⊕ g) ⊕ 1) ⊕ (a ⊕ b)	d	d ⊕ e
CHARACTER SET #2 ACTIVE 0	b ⊕ g	(f ⊕ g) ⊕ (d ⊕ e)	e ⊕ a	a
INPUT CODE	A	B	C	D
CHARACTER SET #1 ACTIVE 1	a ⊙ f	((f ⊙ g) ⊙ 0) ⊙ (a ⊙ b)	d	d ⊙ e
CHARACTER SET #2 ACTIVE 1	b ⊙ g	(f ⊙ g) ⊙ (d ⊙ e)	e ⊙ a	a
CHARACTER SET #1 ACTIVE 0	a ⊙ f	((f ⊙ g) ⊙ 0) ⊙ (a ⊙ b)	d ⊙ 0	d ⊙ e
CHARACTER SET #2 ACTIVE 0	b ⊙ g	(f ⊙ g) ⊙ (d ⊙ e)	e ⊙ a	a ⊙ 0

(⊕ MEANS EXCLUSIVE-OR; ⊙ MEANS EXCLUSIVE-NOR.)